Итоговая работа по курсу

«Программирование на языке С (базовый уровень)»

Петьков Анатолий Валерьевич

Оглавление

[Постановка задания 3](#_heading=h.3znysh7)

[Описание программы 5](#_heading=h.2et92p0)

[Исходный код программы 5](#_heading=h.tyjcwt)

[Описание файлов программы 5](#_heading=h.3dy6vkm)

[Описание функций программы 5](#_heading=h.3dy6vkm)

[Описание работы программы 6](#_heading=h.1t3h5sf)

[Примеры написания командной строки для запуска: 6](#_heading=h.4d34og8)

[Демонстрация работы программы 7](#_heading=h.2s8eyo1)

[Сборка программы утилитой make 8](#_heading=h.17dp8vu)

# Постановка задания

**Создание консольного приложения — Статистика температуры**

Необходимо реализовать консольное приложение, которое осуществляет считывание текстового файла csv, состоящего из строк следующего формата:

**YEAR;MONTH;DAY;HOUR;MINUTE;TEMPERATURE**

dddd;mm;dd;hh;mm;temperature

dddd - год 4 цифры

mm - месяц 2 цифры

dd - день 2 цифры

hh - часы 2 цифры

mm - минуты 2 цифры

temperature - целое число от -99 до 99

В архиве файле хранится статистика собранная датчиком температуры за 1 календарный год. Предполагается, что датчик собирал информацию не чаще чем 1 раз в минуту и сохранял в заданном формате каждое значение в текстовый файл с новой строки.

В какой-то момент времени датчик мог не работать, тогда данные поэтому периоду могут отсутствовать. Пример входного файла:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| YEAR | MONTH | DAY | HOUR | MINUTE | TEMPERATURE |
| 2021 | 1 | 1 | 23 | 1 | -5 |
| 2021 | 1 | 1 | 23 | 3 | -6 |
| 2021 | 1 | 1 | 23 | 1 | -7 |
| 2021 | 1 | 2 | 7 | 5 | -10 |

**Требования к обработке данных**

Необходимо вывести статистику по каждому месяцу, с учетом исходных данных:

* среднемесячная температура
* минимальная температура в текущем месяце
* максимальная температура в текущем месяце

Также необходимо вывести статистику за год:

* среднегодовая температура
* минимальная температура
* максимальная температура

**Требования к аргументам командной строки**

Приложение должно обрабатывать аргументы командной строки:

минимальный набор поддерживаемых ключей:

* -h Описание функционала приложения. Список ключей, которые обрабатывает данное приложение и их назначение.
* -f <filename.csv> входной файл csv для обработки.
* -m <номер месяца> если задан данный ключ, то выводится только статистика за указанный месяц.
* если нет параметров, то выдается help

**Требования к ошибкам в входных данных**

* Приложение должно корректно работать на любых входных данных, если формат csv файла не соответствует заданному, то необходимо указать номер строки файла csv, в которой обнаружена ошибка и не учитывать данную строку.

В архиве с заданием [temperature\_data\_examples.zip](https://drive.google.com/file/d/1BBWwFSzVtGRPgvout8EmWXsdVUEVRtYG/view?usp=drive_link) лежат два файла:

* temperature\_big.csv — файл со статистикой за год
* temperature\_small.csv — укороченный файл с ошибками для тестирования

**Требования к сборке приложения**

* Приложение должно собираться при помощи утилиты make.
* Все прототипы функций, используемые в приложении, должны быть вынесены в отдельный файл temp\_functions.h
* Тексты функций — в файл temp\_functions.c
* Для реализации приложения рекомендуется использовать массив из структурного типа данных для хранения показаний датчика.

# **Описание программы**

## **Исходный код программы:**

|  |
| --- |
| * Ссылка на репозиторий ([Ссылка на репозиторий](https://github.com/Anatolpe/MIPI_BaseC/tree/f58f92d2ba30683fed976a0b2fcad0067b575949/Kursovik)) |

## **Описание файлов программы**

***Программа состоит из следующих файлов***:

***- main.c –*** *основной файл программы, содержащий точку входа (функцию main). Обрабатывает аргументы командной строки, управляет кодировкой консоли и запускает соответствующие функции для обработки данных и вывода статистики.*

***- Temp\_function.h*** *– заголовочный файл. Содержит объявления структур данных (record, data, data\_month), определения констант и прототипы всех функций, используемых для работы с данными.*

***- Temp\_function.c –*** *файл реализации. Содержит код функций для чтения и разбора CSV-файла, проверки данных на корректность, вычисления статистики (среднее, минимум, максимум) и красивого вывода результатов в консоль.*

***- Makefile –*** *файл с инструкциями для утилиты сборки make. Описывает зависимости между файлами и команды для компиляции и линковки финального исполняемого файла.*

***- build.bat –*** *файл автоматической сборки и компиляции всего проекта.*

***- temperature\_stat.exe–*** *исполняемый файл, результат успешной сборки программы.*

***Описание функций программы***

Для лучшего понимания архитектуры программы ниже приведено описание основных функций, сгруппированных по файлам.

**Файл main.c**

Содержит точку входа и функции для взаимодействия с пользователем.

Функции:

main(int32\_t argc, char \*argv[]) - Точка входа в программу. Устанавливает кодировку консоли UTF-8 для корректного отображения и вызывает функцию обработки аргументов args.

args(int32\_t argc, char \*argv[]) - Обрабатывает аргументы командной строки. Анализирует переданные ключи (-f, -m, -h, -v) и управляет логикой выполнения программы (вывод справки, вызов обработки файла).

process\_file\_and\_show\_stats(char \*filename, uint8\_t month) - Центральная функция управления. Вызывает read\_data для загрузки файла, а затем, в зависимости от параметра month, вызывает либо print\_month\_info (если месяц указан), либо print\_yearstat\_info (для вывода всей статистики).

print\_help(void) - Выводит справку по использованию программы.

print\_about(void) - Выводит информацию о программе при запуске без аргументов.

**Файл Temp\_function.h**

Содержит объявления структур и прототипы функций.

Структуры данных:

record - хранит одну запись о температуре (год, месяц, день, час, минута, температура).

data - массив записей record для хранения всех данных, загруженных из файла.

data\_month - массив записей record для хранения данных, отфильтрованных по одному месяцу.

Файл Temp\_function.c

Содержит реализацию логики работы с данными.

Функции для работы с данными:

read\_data(data \*dt, char \*csv\_name) - Основная функция чтения CSV-файла. Последовательно считывает строки, пропускает заголовок, парсит данные с помощью char2num, выполняет валидацию всех полей (проверяет диапазоны значений, ищет дубликаты). Корректные записи добавляются в массив структур.

char2num(char \*p, char stop\_symb) - Вспомогательная функция для преобразования строки в число. Обрабатывает отрицательные числа и пробелы. Возвращает DATA\_ERROR при обнаружении недопустимых символов.

add\_record(...) - Вспомогательная функция для заполнения полей структуры record данными.

Функции для расчета статистики:

get\_month\_data(data \*dt\_source, data\_month \*dt\_dest, uint8\_t month\_number) - Фильтрует данные из общего массива dt\_source, выбирая только записи за указанный месяц, и помещает их в dt\_dest.

month\_average\_temp(data\_month \*dt) - Вычисляет среднюю температуру по всем записям в структуре данных месяца.

sortByTemp(data\_month \*dt) - Сортирует массив записй в структуре данных месяца по температуре по возрастанию. После сортировки первый элемент массива является минимальной температурой, а последний — максимальной.

Функции для вывода информации:

print\_month\_info(data \*dt\_source, data\_month \*dt\_dest, uint8\_t num\_month) - Выводит статистику (среднюю, мин., макс. температуру) для одного месяца. Вызывает get\_month\_data, month\_average\_temp и sortByTemp.

print\_yearstat\_info(data \*dt\_source, data\_month \*dt\_dest) - Выводит полную статистику за год. В цикле обрабатывает каждый месяц, вычисляет его статистику, а на основе результатов вычисляет и выводит общую годовую статистику.

print\_month\_data(), print\_year\_data() - Вспомогательные функции для форматированного вывода заголовков и данных таблиц.

## **Описание работы программы**

Программа представляет собой консольное приложение для анализа температурных данных.

Основные возможности:

* Чтение данных из CSV-файлов
* Валидация входных данных
* Расчет статистических показателей
* Вывод результатов в консоль

Рекомендуется запускать её из командной строки.

При запуске без указания ключей, программа кратко выведет информацию о своем назначении и предложении ввести ключ «-h» для получения инструкций.

Программа допускает применение следующих ключей:

* «-h» - получение информации о возможных ключах запуска с кратким описанием их назначения;
* «-f file\_name» - указание файла для обработки, где file\_name – имя файла. Если в этом режиме не добавлен ключ «-m», то выведется полная статистика по всему файлу;
* «-m xx» - указание месяца для получения статистики по конкретному месяцу, где xx - месяц;
* «-v» - показывает версию программного обеспечения.

### Примеры написания командной строки для запуска:

temperature\_stat.exe # Вывод справки

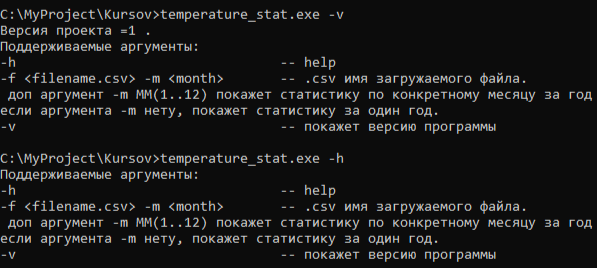
temperature\_stat.exe -h # Подробная справка

temperature\_stat.exe -f data.csv # Анализ всего файла

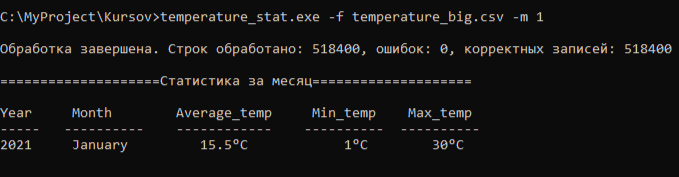
temperature\_stat.exe -f data.csv -m 3 # Анализ марта

### **Демонстрация работы программы**

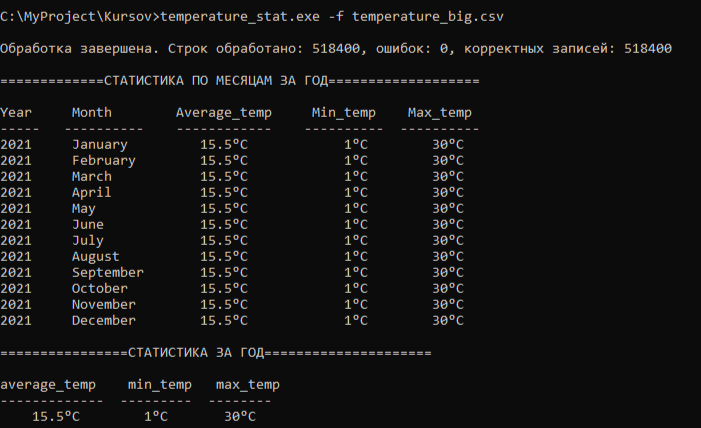
**Рисунок 1**. **Вывод справки программы**



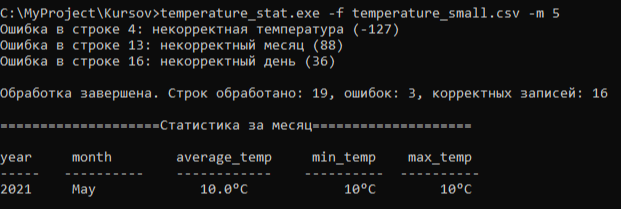
**Рисунок 2**. **Запуск программы с указанием файла и опцией выбора месяца:**



**Рисунок 3**. **Запуск программы с указанием имени файла:**

****

**Рисунок 4**. **Запуск программы с указанием имени файла содержащего разные ошибки:**



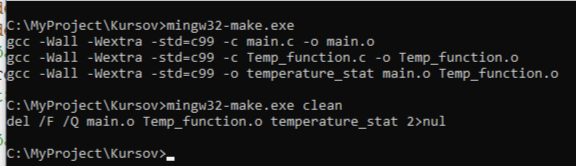
### **Сборка программы утилитой make**

### Сборка программы

Для сборки используется утилита **make**:

mingw32-make.exe # Сборка программы

mingw32-make.exe clean # Очистка проекта

****

**Требования к окружению**:

* Компилятор GCC
* Утилита make
* Стандартная библиотека C

Программа кроссплатформенная и может быть собрана на любой системе с поддержкой GCC.